新しくなった4Dのジャーナルシステム

original presentation by Laurent Ribardière

ロホン・リバルディエール

ジャーナルファイル新フォーマット

ヘッダー（4バイト）
オペレーション番号（8バイト）
サイズ（4バイト）
オペレーションタイプ（4バイト）
オペレーションタイプに依拠する内容: レコード値など
サイズ（繰り返し）（4バイト）
フッター（4バイト）

主要な変更点

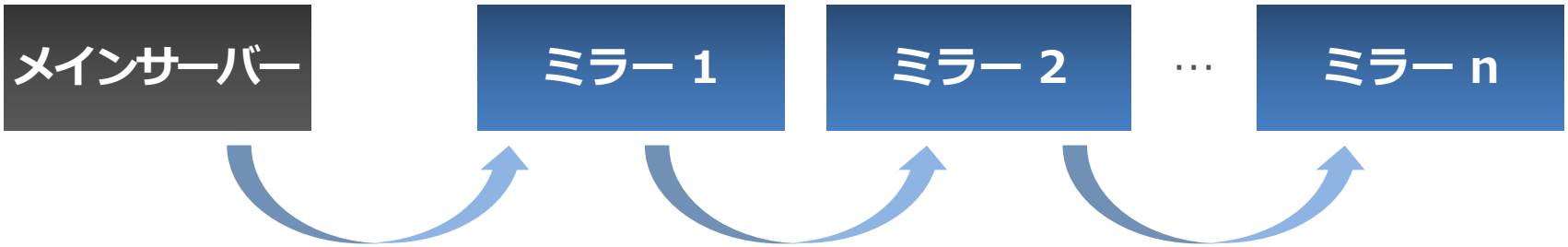
1. グルーバル・オペレーション・カウンターを進めるのは“データを変更するオペレーション” のみ
2. レコード番号の代わりにプライマリーキーを識別子に使用
3. トランザクションがジャーナルに記録されるのは確定後
4. 逆方向にナビゲーションできるジャーナルのフォーマット
5. ジャーナルに記録されるテーブルは選択できる
6. 別データファイルのジャーナルも統合できる

ポイント❶

グルーバル・オペレーション・カウンターを進めるのは
“データを変更するオペレーション” のみ

1.手動および連鎖ミラーリングが容易

2.連鎖: メインサーバーの負荷を軽減



カウンターをインクリメントするオペレーションを限定

旧方式

記録スタート	*1*
データベースを開く	*2*
データベースを閉じる	*3*
データベースを開く	*4*
レコードを変更	*5*
レコードを追加	*6*
レコードを追加	*7*
データベースを閉じる	*8*

新方式

記録スタート	*1*
データベースを開く	*1*
データベースを閉じる	*1*
データベースを開く	*1*
レコードを変更	*2*
レコードを追加	*3*
レコードを追加	*4*
データベースを閉じる	*4*

ポイント❷

レコード番号の代わりにプライマリーキーを識別子に使用

1. 堅牢性

2. ジャーナルが破損していたとしても，
   部分的な復元ができる

レコード番号の代わりにプライマリーキーを識別子に使用

旧方式

1 - レコード作成:
フィールド値 {a, b, c}，レコード番号 #10

2 - レコード作成:
フィールド値 {d, e, f}，レコード番号 #11

3 - レコード作成:
フィールド値 {g, h, i}，レコード番号 #12

4 - レコード更新:
レコード番号 #10，値 {a2, b2, c2}

5 - レコード更新:
レコード番号 #11，値 {d2, e2, f2}

新方式

1. レコード作成:
フィールド値 {a, b, c}，主キー [x]

2 - レコード作成:
フィールド値 {d, e, f}，主キー [y]

3 - レコード作成:
フィールド値 {g, h, i}，主キー [z]

4 - レコード更新:
主キー [x] ，値{a2, b2, c2}

5 - レコード更新:
主キー [y] ，値{d2, e2, f2}

ポイント❸

トランザクションがジャーナルに記録されるのは確定後

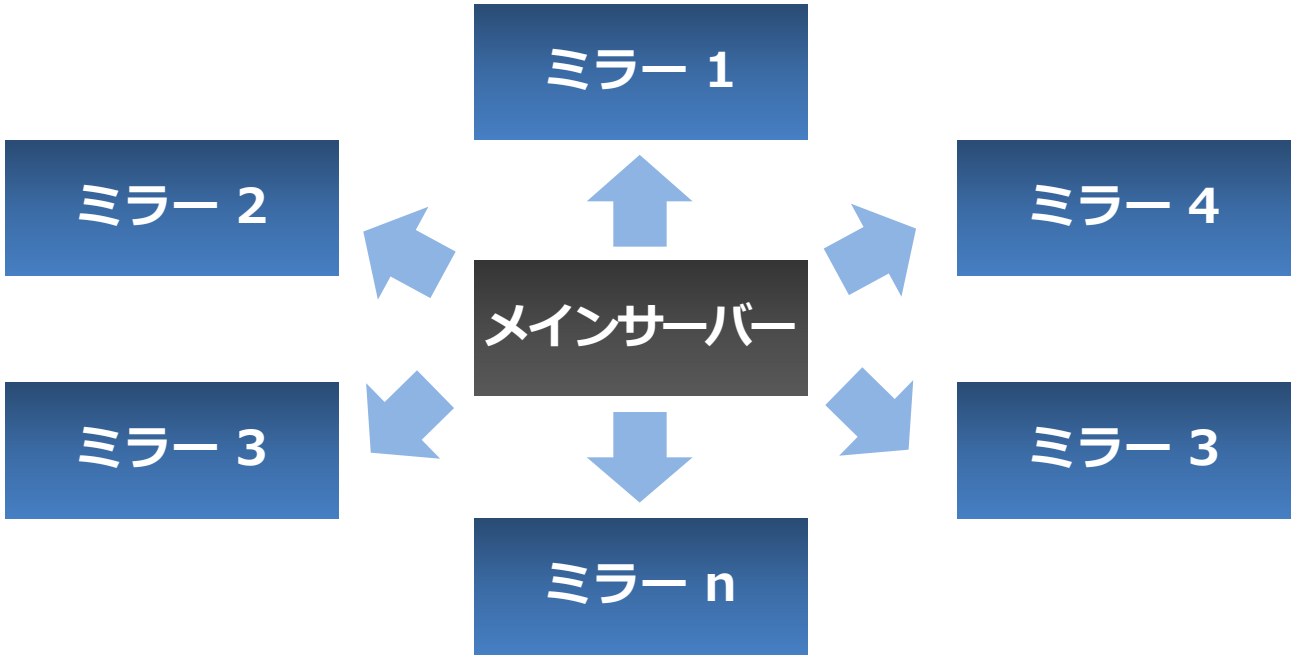
1. 少ない分量でミラーリングを実行

2. ジャーナルの不要な肥大化を防止

ポイント❹

逆方向にナビゲーションできるジャーナルのフォーマット

1.分散ミラーリング



ポイント❺

ジャーナルに記録されるテーブルは選択できる

1. 一時テーブルの処理を高速に

ポイント❻

別データファイルのジャーナルも統合できる

1.手動ミラーリングが容易

2.データベースのバックアップを
　全然，実行しなくても構わない